My Project

Generated by Doxygen 1.10.0

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Asema Class Reference

`#include <asema.h>`

Public Member Functions

- Asema ()

  Asema::Asema.
- void paivitaAsema (Siirto ∗)
- double evaluoi ()
- MinMaxPaluu maxi (int syvyys, Asema ∗a, double alpha, double beta)
- MinMaxPaluu mini (int syvyys, Asema ∗a, double alpha, double beta)
- MinMaxPaluu minimax (int syvyys)
- void annaLaillisetSiirrot (std::list< Siirto > &lista)
- int getSiirtovuoro ()
- void setSiirtovuoro (int)
- bool getOnkoValkeaKuningasLiikkunut ()
- bool getOnkoMustaKuningasLiikkunut ()
- bool getOnkoValkeaDTliikkunut ()
- bool getOnkoValkeaKTliikkunut ()
- bool getOnkoMustaDTliikkunut ()
- bool getOnkoMustaKTliikkunut ()
- void annaVastustajanSiirrot (std::list< Siirto > &lista, int vastustajanVari)
- double kuningasTurvassa (int vari)

Public Attributes

- Nappula ∗ _lauta [8][8]
- int kaksoisaskelSarakkeella = -1
- bool ihmisenVuoro
- Ruutu valkeanKuninkaanRuutu
- Ruutu mustanKuninkaanRuutu

Static Public Attributes

- static Nappula * vk = new Kuningas(L"\u265A", 0, VK)
- static Nappula * vd = new Daami(L"\u265B", 0, VD)
- static Nappula * vt = new Torni(L"\u265C", 0, VT)
- static Nappula * vl = new Lahetti(L"\u265D", 0, VL)
- static Nappula * vr = new Ratsu(L"\u265E", 0, VR)
- static Nappula * vs = new Sotilas(L"\u265F", 0, VS)
- static Nappula * mk = new Kuningas(L"\u265A", 1, MK)
- static Nappula * md = new Daami(L"\u265B", 1, MD)
- static Nappula * mt = new Torni(L"\u265C", 1, MT)
- static Nappula * ml = new Lahetti(L"\u265D", 1, ML)
- static Nappula * mr = new Ratsu(L"\u265E", 1, MR)
- static Nappula * ms = new Sotilas(L"\u265F", 1, MS)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 Asema()

```
Asema::Asema ( )
```

Asema::Asema.

Returns

Asema

4.1.2 Member Function Documentation

4.1.2.1 annaLaillisetSiirrot()

```
void Asema::annaLaillisetSiirrot (
            std::list< Siirto > & lista )
```

4.1.2.2 annaVastustajanSiirrot()

```
void Asema::annaVastustajanSiirrot (
            std::list< Siirto > & lista,
            int vastustajanVari )
```

4.1.2.3 evaluoi()

```
double Asema::evaluoi ( )
```

4.1.2.4 getOnkoMustaDTliikkunut()

```
bool Asema::getOnkoMustaDTliikkunut ( )
```

4.1.2.5 getOnkoMustaKTliikkunut()

```
bool Asema::getOnkoMustaKTliikkunut ( )
```

4.1.2.6 getOnkoMustaKuningasLiikkunut()

```
bool Asema::getOnkoMustaKuningasLiikkunut ( )
```

4.1.2.7 getOnkoValkeaDTliikkunut()

```
bool Asema::getOnkoValkeaDTliikkunut ( )
```

4.1.2.8 getOnkoValkeaKTliikkunut()

```
bool Asema::getOnkoValkeaKTliikkunut ( )
```

4.1.2.9 getOnkoValkeaKuningasLiikkunut()

```
bool Asema::getOnkoValkeaKuningasLiikkunut ( )
```

4.1.2.10 getSiirtovuoro()

```
int Asema::getSiirtovuoro ( )
```

4.1.2.11 kuningasTurvassa()

```
double Asema::kuningasTurvassa (
            int vari )
```

4.1.2.12 maxi()

```
MinMaxPaluu Asema::maxi (
            int syvyys,
            Asema * a,
            double alpha,
            double beta )
```

4.1.2.13 mini()

```
MinMaxPaluu Asema::mini (
            int syvyys,
            Asema * a,
            double alpha,
            double beta )
```

4.1.2.14 minimax()

```
MinMaxPaluu Asema::minimax (
            int syvyys )
```

4.1.2.15 paivitaAsema()

```
void Asema::paivitaAsema (
            Siirto * siirto )
```

4.1.2.16 setSiirtovuoro()

```
void Asema::setSiirtovuoro (
            int vuoro )
```

4.1.3 Member Data Documentation

4.1.3.1 _lauta

```
Nappula* Asema::_lauta[8][8]
```

4.1.3.2 ihmisenVuoro

```
bool Asema::ihmisenVuoro
```

4.1.3.3 kaksoisaskelSarakkeella

```
int Asema::kaksoisaskelSarakkeella = -1
```

4.1.3.4 md

```
Nappula * Asema::md = new Daami(L"\u265B", 1, MD)  [static]
```

4.1.3.5 mk

```
Nappula * Asema::mk = new Kuningas(L"\u265A", 1, MK)  [static]
```

4.1.3.6 ml

```
Nappula * Asema::ml = new Lahetti(L"\u265D", 1, ML)  [static]
```

4.1.3.7 mr

```
Nappula * Asema::mr = new Ratsu(L"\u265E", 1, MR)  [static]
```

4.1.3.8 ms

```
Nappula * Asema::ms = new Sotilas(L"\u265F", 1, MS)  [static]
```

4.1.3.9 mt

```
Nappula * Asema::mt = new Torni(L"\u265C", 1, MT)  [static]
```

4.1.3.10 mustanKuninkaanRuutu

```
Ruutu Asema::mustanKuninkaanRuutu
```

4.1.3.11 valkeanKuninkaanRuutu

```
Ruutu Asema::valkeanKuninkaanRuutu
```

4.1.3.12 vd

```
Nappula * Asema::vd = new Daami(L"\u265B", 0, VD)  [static]
```

4.1.3.13 vk

```
Nappula * Asema::vk = new Kuningas(L"\u265A", 0, VK)  [static]
```

4.1.3.14 vl

```
Nappula * Asema::vl = new Lahetti(L"\u265D", 0, VL)  [static]
```

4.1.3.15 vr

```
Nappula * Asema::vr = new Ratsu(L"\u265E", 0, VR)  [static]
```

4.1.3.16 vs

```
Nappula * Asema::vs = new Sotilas(L"\u265F", 0, VS)  [static]
```

4.1.3.17 vt

```
Nappula * Asema::vt = new Torni(L"\u265C", 0, VT)  [static]
```

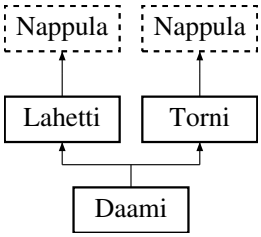
The documentation for this class was generated from the following files:

- shakki/asema.h
- shakki/asema.cpp

4.2 Daami Class Reference

```
#include <nappula.h>
```

Inheritance diagram for Daami:



Public Member Functions

- Daami (std::wstring unicode, int vari, int koodi)
- void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)

Public Member Functions inherited from Lahetti

- Lahetti (std::wstring unicode, int vari, int koodi)
- void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)

Public Member Functions inherited from Nappula

- Nappula (std::wstring, int, int)
- Nappula ()
- void setUnicode (std::wstring unicode)
- std::wstring getUnicode ()
- void setVari (int vari)
- int getVari ()
- int getKoodi ()
- void setKoodi (int koodi)
- void siirrotSuuntaan (std::list< Siirto > &lista, Ruutu *, Asema *, int vari, int dx, int dy, int askeleet)
- void lisaaSotilaanKorotukset (Siirto *, std::list< Siirto > &lista, Asema *)

Public Member Functions inherited from Torni

- Torni (std::wstring unicode, int vari, int koodi)
- void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Daami()

```
Daami::Daami (
            std::wstring unicode,
            int vari,
            int koodi )  [inline]
```

4.2.2 Member Function Documentation

4.2.2.1 annaSiirrot()

```
void Daami::annaSiirrot (
            std::list< Siirto > & lista,
            Ruutu * ruutu,
            Asema * asema,
            int vari )  [virtual]
```

Implements Nappula.

The documentation for this class was generated from the following files:

- shakki/nappula.h
- shakki/nappula.cpp

4.3 Kayttoliittyma Class Reference

```
#include <kayttoliittyma.h>
```

Public Member Functions

- void aseta_asema (Asema *asema)
- void piirraLauta (std::list< Siirto > &lista)

  Aseta asema.
- Siirto annaVastustajanSiirto (std::list< Siirto > &lista)
- int kysyVastustajanVari ()

Static Public Member Functions

- static Kayttoliittyma * getInstance ()

4.3.1 Member Function Documentation

4.3.1.1 annaVastustajanSiirto()

```
Siirto Kayttoliittyma::annaVastustajanSiirto (
            std::list< Siirto > & lista )
```

4.3.1.2 aseta_asema()

```
void Kayttoliittyma::aseta_asema (
            Asema * asema )  [inline]
```

4.3.1.3 getInstance()

```
Kayttoliittyma * Kayttoliittyma::getInstance ( )  [static]
```

4.3.1.4 kysyVastustajanVari()

```
int Kayttoliittyma::kysyVastustajanVari ( )
```

4.3.1.5 piirraLauta()

```
void Kayttoliittyma::piirraLauta (
            std::list< Siirto > & lista )
```

Aseta asema.

Parameters

asema	

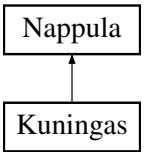
The documentation for this class was generated from the following files:

- shakki/kayttoliittyma.h
- shakki/kayttoliittyma.cpp

4.4 Kuningas Class Reference

`#include <nappula.h>`

Inheritance diagram for Kuningas:



Public Member Functions

- Kuningas (std::wstring unicode, int vari, int koodi)
- void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)

Public Member Functions inherited from Nappula

- Nappula (std::wstring, int, int)
- Nappula ()
- void setUnicode (std::wstring unicode)
- std::wstring getUnicode ()
- void setVari (int vari)
- int getVari ()
- int getKoodi ()
- void setKoodi (int koodi)
- void siirrotSuuntaan (std::list< Siirto > &lista, Ruutu *, Asema *, int vari, int dx, int dy, int askeleet)
- void lisaaSotilaanKorotukset (Siirto *, std::list< Siirto > &lista, Asema *)

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Kuningas()

```
Kuningas::Kuningas (
            std::wstring unicode,
            int vari,
            int koodi )  [inline]
```

4.4.2 Member Function Documentation

4.4.2.1 annaSiirrot()

```
void Kuningas::annaSiirrot (
            std::list< Siirto > & lista,
            Ruutu * ruutu,
            Asema * asema,
            int vari )  [virtual]
```

Implements Nappula.

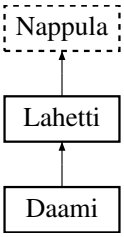
The documentation for this class was generated from the following files:

- shakki/nappula.h
- shakki/nappula.cpp

4.5 Lahetti Class Reference

`#include <nappula.h>`

Inheritance diagram for Lahetti:



Public Member Functions

- Lahetti (std::wstring unicode, int vari, int koodi)
- void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)

Public Member Functions inherited from Nappula

- Nappula (std::wstring, int, int)
- Nappula ()
- void setUnicode (std::wstring unicode)
- std::wstring getUnicode ()
- void setVari (int vari)
- int getVari ()
- int getKoodi ()
- void setKoodi (int koodi)
- void siirrotSuuntaan (std::list< Siirto > &lista, Ruutu *, Asema *, int vari, int dx, int dy, int askeleet)
- void lisaaSotilaanKorotukset (Siirto *, std::list< Siirto > &lista, Asema *)

4.5.1 Constructor & Destructor Documentation

4.5.1.1 Lahetti()

```
Lahetti::Lahetti (
            std::wstring unicode,
            int vari,
            int koodi )  [inline]
```

4.5.2 Member Function Documentation

4.5.2.1 annaSiirrot()

```
void Lahetti::annaSiirrot (
            std::list< Siirto > & lista,
            Ruutu * ruutu,
            Asema * asema,
            int vari )  [virtual]
```

Implements Nappula.

The documentation for this class was generated from the following files:

- shakki/nappula.h
- shakki/nappula.cpp

4.6 MinMaxPaluu Class Reference

`#include <minmaxpaluu.h>`

Public Attributes

- double _evaluointiArvo
- Siirto _parasSiirto

4.6.1 Member Data Documentation

4.6.1.1 _evaluointiArvo

`double MinMaxPaluu::_evaluointiArvo`

4.6.1.2 _parasSiirto

`Siirto MinMaxPaluu::_parasSiirto`

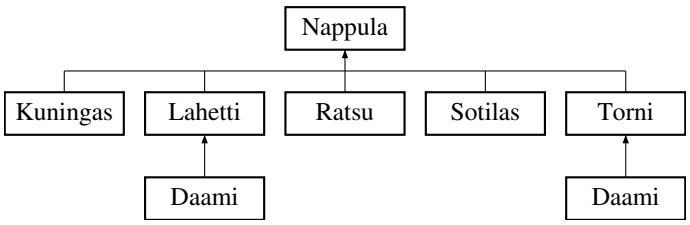
The documentation for this class was generated from the following file:

- shakki/minmaxpaluu.h

4.7 Nappula Class Reference

`#include <nappula.h>`

Inheritance diagram for Nappula:



Public Member Functions

- Nappula (std::wstring, int, int)
- Nappula ()
- virtual void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)=0
- void setUnicode (std::wstring unicode)
- std::wstring getUnicode ()
- void setVari (int vari)
- int getVari ()
- int getKoodi ()
- void setKoodi (int koodi)
- void siirrotSuuntaan (std::list< Siirto > &lista, Ruutu *, Asema *, int vari, int dx, int dy, int askeleet)
- void lisaaSotilaanKorotukset (Siirto *, std::list< Siirto > &lista, Asema *)

4.7.1 Constructor & Destructor Documentation

4.7.1.1 Nappula() [1/2]

```
Nappula::Nappula (
            std::wstring ,
            int ,
            int  )
```

4.7.1.2 Nappula() [2/2]

```
Nappula::Nappula ( )  [inline]
```

4.7.2 Member Function Documentation

4.7.2.1 annaSiirrot()

```
virtual void Nappula::annaSiirrot (
            std::list< Siirto > & lista,
            Ruutu * ,
            Asema * ,
            int vari )  [pure virtual]
```

Implemented in Torni, Ratsu, Lahetti, Daami, Kuningas, and Sotilas.

4.7.2.2 getKoodi()

```
int Nappula::getKoodi ( )  [inline]
```

4.7.2.3 getUnicode()

```
std::wstring Nappula::getUnicode ( )  [inline]
```

4.7.2.4 getVari()

```
int Nappula::getVari ( )  [inline]
```

4.7.2.5 lisaaSotilaanKorotukset()

```
void Nappula::lisaaSotilaanKorotukset (
            Siirto * siirto,
            std::list< Siirto > & lista,
            Asema * asema )
```

4.7.2.6 setKoodi()

```
void Nappula::setKoodi (
            int koodi )  [inline]
```

4.7.2.7 setUnicode()

```
void Nappula::setUnicode (
            std::wstring unicode )  [inline]
```

4.7.2.8 setVari()

```
void Nappula::setVari (
            int vari )  [inline]
```

4.7.2.9 siirrotSuuntaan()

```
void Nappula::siirrotSuuntaan (
            std::list< Siirto > & lista,
            Ruutu * ruutu,
            Asema * asema,
            int vari,
            int dx,
            int dy,
            int askeleet )
```

The documentation for this class was generated from the following files:

- shakki/nappula.h
- shakki/nappula.cpp

4.8 Peli Class Reference

```
#include <peli.h>
```

Public Member Functions

- Peli (int)
- int getKoneenVari ()

4.8.1 Constructor & Destructor Documentation

4.8.1.1 Peli()

```
Peli::Peli (
            int ihmisenVari )
```

4.8.2 Member Function Documentation

4.8.2.1 getKoneenVari()

```
int Peli::getKoneenVari ( )
```

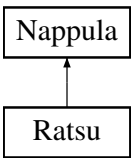
The documentation for this class was generated from the following files:

- shakki/peli.h
- shakki/peli.cpp

4.9 Ratsu Class Reference

```
#include <nappula.h>
```

Inheritance diagram for Ratsu:



Public Member Functions

- Ratsu (std::wstring unicode, int vari, int koodi)
- void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)

Public Member Functions inherited from Nappula

- Nappula (std::wstring, int, int)
- Nappula ()
- void setUnicode (std::wstring unicode)
- std::wstring getUnicode ()
- void setVari (int vari)
- int getVari ()
- int getKoodi ()
- void setKoodi (int koodi)
- void siirrotSuuntaan (std::list< Siirto > &lista, Ruutu *, Asema *, int vari, int dx, int dy, int askeleet)
- void lisaaSotilaanKorotukset (Siirto *, std::list< Siirto > &lista, Asema *)

4.9.1 Constructor & Destructor Documentation

4.9.1.1 Ratsu()

```
Ratsu::Ratsu (
            std::wstring unicode,
            int vari,
            int koodi )  [inline]
```

4.9.2 Member Function Documentation

4.9.2.1 annaSiirrot()

```
void Ratsu::annaSiirrot (
            std::list< Siirto > & lista,
            Ruutu * ruutu,
            Asema * asema,
            int vari )  [virtual]
```

Implements Nappula.

The documentation for this class was generated from the following files:

- shakki/nappula.h
- shakki/nappula.cpp

4.10 Ruutu Class Reference

```
#include <ruutu.h>
```

Public Member Functions

- Ruutu (int, int)
- Ruutu ()
- bool operator== (const Ruutu &) const
- int getRivi ()
- int getSarake ()
- void setRivi (int)
- void setSarake (int)

4.10.1 Constructor & Destructor Documentation

4.10.1.1 Ruutu() [1/2]

```
Ruutu::Ruutu (
            int sarake,
            int rivi )
```

4.10.1.2 Ruutu() [2/2]

```
Ruutu::Ruutu ( )
```

4.10.2 Member Function Documentation

4.10.2.1 getRivi()

```
int Ruutu::getRivi ( )
```

4.10.2.2 getSarake()

```
int Ruutu::getSarake ( )
```

4.10.2.3 operator==()

```
bool Ruutu::operator== (
            const Ruutu & ruutu ) const
```

4.10.2.4 setRivi()

```
void Ruutu::setRivi (
            int rivi )
```

4.10.2.5 setSarake()

```
void Ruutu::setSarake (
            int sarake )
```

The documentation for this class was generated from the following files:

- shakki/ruutu.h
- shakki/ruutu.cpp

4.11 Siirto Class Reference

```
#include <siirto.h>
```

Public Member Functions

- Siirto (Ruutu, Ruutu)
- Siirto ()
- Siirto (bool, bool)
- bool operator== (const Siirto &) const
- Ruutu getAlkuruutu ()
- Ruutu getLoppuruutu ()
- bool onkoLyhytLinna ()
- bool onkoPitkalinna ()
- void setAlkuruutu (Ruutu)
- void setLoppuruutu (Ruutu)

Public Attributes

- Nappula * _miksikorotetaan = 0

4.11.1 Constructor & Destructor Documentation

4.11.1.1 Siirto() [1/3]

```
Siirto::Siirto (
            Ruutu alkuRuutu,
            Ruutu loppuRuutu )
```

4.11.1.2 Siirto() [2/3]

```
Siirto::Siirto ( )
```

4.11.1.3 Siirto() [3/3]

```
Siirto::Siirto (
            bool lyhytLinna,
            bool pitkaLinna )
```

4.11.2 Member Function Documentation

4.11.2.1 getAlkuruutu()

```
Ruutu Siirto::getAlkuruutu ( )
```

4.11.2.2 getLoppuruutu()

```
Ruutu Siirto::getLoppuruutu ( )
```

4.11.2.3 onkoLyhytLinna()

```
bool Siirto::onkoLyhytLinna ( )
```

4.11.2.4 onkoPitkalinna()

```
bool Siirto::onkoPitkalinna ( )
```

4.11.2.5 operator==()

```
bool Siirto::operator== (
            const Siirto & siirto ) const
```

4.11.2.6 setAlkuruutu()

```
void Siirto::setAlkuruutu (
            Ruutu alkuRuutu )
```

4.11.2.7 setLoppuruutu()

```
void Siirto::setLoppuruutu (
            Ruutu loppuRuutu )
```

4.11.3 Member Data Documentation

4.11.3.1 _miksikorotetaan

```
Nappula* Siirto::_miksikorotetaan = 0
```

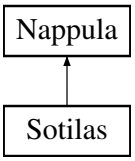
The documentation for this class was generated from the following files:

- shakki/siirto.h
- shakki/siirto.cpp

4.12 Sotilas Class Reference

```
#include <nappula.h>
```

Inheritance diagram for Sotilas:



Public Member Functions

- Sotilas (std::wstring unicode, int vari, int koodi)
- void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)

Public Member Functions inherited from Nappula

- Nappula (std::wstring, int, int)
- Nappula ()
- void setUnicode (std::wstring unicode)
- std::wstring getUnicode ()
- void setVari (int vari)
- int getVari ()
- int getKoodi ()
- void setKoodi (int koodi)
- void siirrotSuuntaan (std::list< Siirto > &lista, Ruutu *, Asema *, int vari, int dx, int dy, int askeleet)
- void lisaaSotilaanKorotukset (Siirto *, std::list< Siirto > &lista, Asema *)

4.12.1 Constructor & Destructor Documentation

4.12.1.1 Sotilas()

```
Sotilas::Sotilas (
            std::wstring unicode,
            int vari,
            int koodi )  [inline]
```

4.12.2 Member Function Documentation

4.12.2.1 annaSiirrot()

```
void Sotilas::annaSiirrot (
            std::list< Siirto > & lista,
            Ruutu * ruutu,
            Asema * asema,
            int vari )  [virtual]
```

Implements Nappula.

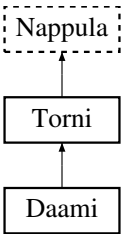
The documentation for this class was generated from the following files:

- shakki/nappula.h
- shakki/nappula.cpp

4.13 Torni Class Reference

```
#include <nappula.h>
```

Inheritance diagram for Torni:



Public Member Functions

- Torni (std::wstring unicode, int vari, int koodi)
- void annaSiirrot (std::list< Siirto > &lista, Ruutu *, Asema *, int vari)

Public Member Functions inherited from Nappula

- Nappula (std::wstring, int, int)
- Nappula ()
- void setUnicode (std::wstring unicode)
- std::wstring getUnicode ()
- void setVari (int vari)
- int getVari ()
- int getKoodi ()
- void setKoodi (int koodi)
- void siirrotSuuntaan (std::list< Siirto > &lista, Ruutu *, Asema *, int vari, int dx, int dy, int askeleet)
- void lisaaSotilaanKorotukset (Siirto *, std::list< Siirto > &lista, Asema *)

4.13.1 Constructor & Destructor Documentation

4.13.1.1 Torni()

```
Torni::Torni (
            std::wstring unicode,
            int vari,
            int koodi )  [inline]
```

4.13.2 Member Function Documentation

4.13.2.1 annaSiirrot()

```
void Torni::annaSiirrot (
            std::list< Siirto > & lista,
            Ruutu * ruutu,
            Asema * asema,
            int vari )  [virtual]
```

Implements Nappula.

The documentation for this class was generated from the following files:

- shakki/nappula.h
- shakki/nappula.cpp

4.14 Vastustaja Class Reference

```
#include <vastustaja.h>
```

Public Member Functions

- Vastustaja (std::wstring)
- std::wstring getNimimerkki ()

4.14.1 Constructor & Destructor Documentation

4.14.1.1 Vastustaja()

```
Vastustaja::Vastustaja (
            std::wstring  )
```

4.14.2 Member Function Documentation

4.14.2.1 getNimimerkki()

```
wstring Vastustaja::getNimimerkki ( )
```

The documentation for this class was generated from the following files:

- shakki/vastustaja.h
- shakki/vastustaja.cpp

Chapter 5

File Documentation

5.1 shakki/asema.cpp File Reference

```
#include "asema.h"
#include <chrono>
#include <iostream>
#include "minmaxpaluu.h"
#include "nappula.h"
#include "ruutu.h"
```

5.2 shakki/asema.h File Reference

```
#include <list>
#include <string>
#include "minmaxpaluu.h"
#include "siirto.h"
```

Classes

- class Asema

5.3 asema.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include <list>
00004 #include <string>
00005 #include "minmaxpaluu.h"
00006 #include "siirto.h"
00007
00008 // Ns. "forward declaration". Nyt Asema-luokassa voidaa esitell\x00f6 Nappula-osoittimia ilman,
00009 // ett\x00f6 nappula.h -tiedostoa t\x00f6ytyy includoida.
00010 class Nappula;
00011
00012
```

```
00013 // Asema sis♦lt♦♦ kaiken tarvittavan informaation pelitilanteen kuvaamiseksi
00014 // (nappuloiden sijainti, siirtovuoro, linnoitusoikeudet jne.).
00015 class Asema
00016 {
00017
00018 public:
00019     // Pelilauta sis♦lt♦♦ osoittimet kunkin ruudun nappula-olioon (nullptr/NULL/0 jos ruutu on tyhj♦).
00020     // Public-m♦♦reell♦, koska t♦t♦ k♦ytet♦♦n paljon muualla.
00021     Nappula* _lauta[8][8];
00022
00023     // Nappula-oliot. Huomaa, ett♦ samaa nappulaa voidaan k♦ytt♦♦ useissa eri ruuduissa.
00024     // M♦♦ritelty static-m♦♦reell♦, joten nappulat ovat kaikkien lauta-olioiden "yhteisk♦yt♦ss♦"
00025     // (suorituskyvyn vuoksi).
00026     static Nappula *vk, *vd, *vt, *vl, *vr, *vs; // Valkeat nappulat.
00027     static Nappula *mk, *md, *mt, *ml, *mr, *ms; // Mustat nappulat.
00028
00029     // Ohestaly♦nti♦ varten (-1 = sotilaan kaksoisaskelta ei tapahtunut edellisell♦ siirrolla).
00030     int kaksoisaskelSarakkeella = -1;
00031
00032
00033     Asema();                                              // Asettaa alkuaseman.
00034     void paivitaAsema(Siirto*);                             // P♦ivitt♦♦ aseman annetulla siirrolla.
00035     double evaluoi();                                      // Aseman numeerinen arviointi.
00036     MinMaxPaluu maxi(int syvyys, Asema* a, double alpha, double beta);                          //
      Minimax (max:n siirtovuoro).
00037     MinMaxPaluu mini(int syvyys, Asema* a, double alpha, double beta);                          //
      Minimax (min:n siirtovuoro).
00038     MinMaxPaluu minimax(int syvyys);                      // Minimax-algoritmi.
00039     void annaLaillisetSiirrot(std::list<Siirto>& lista);    // Siirtogeneraattori.
00040     int getSiirtovuoro();                                 // Palauttaa siirtovuoron.
00041     void setSiirtovuoro(int);                             // Asettaa siirtovuoron.
00042     bool getOnkoValkeaKuningasLiikkunut();                    // Linnoittuminen mahdollista?
00043     bool getOnkoMustaKuningasLiikkunut();                   // Linnoittuminen mahdollista?
00044     bool getOnkoValkeaDTliikkunut();                        // Linnoittuminen mahdollista?
00045     bool getOnkoValkeaKTliikkunut();                        // Linnoittuminen mahdollista?
00046     bool getOnkoMustaDTliikkunut();                         // Linnoittuminen mahdollista?
00047     bool getOnkoMustaKTliikkunut();                         // Linnoittuminen mahdollista?
00048     void annaVastustajanSiirrot(std::list<Siirto>& lista, int vastustajanVari);
00049     double kuningasTurvassa(int vari);
00050
00051     bool ihmisenVuoro;
00052     Ruutu valkeanKuninkaanRuutu;
00053     Ruutu mustanKuninkaanRuutu;
00054
00055 private:
00056
00057     // Lis♦informaatio pelitilanteesta.
00058     int _siirtovuoro;                   // 0 = valkea, 1 = musta.
00059     bool _onkoValkeaKuningasLiikkunut;  // Linnoitus ei ole sallittu, jos kuningas on liikkunut.
00060     bool _onkoMustaKuningasLiikkunut;   // Linnoitus ei ole sallittu, jos kuningas on liikkunut.
00061     bool _onkoValkeaDTliikkunut;        // Linnoitus ei ole sallittu, jos daamisivustan torni on
      liikkunut.
00062     bool _onkoValkeaKTliikkunut;        // Linnoitus ei ole sallittu, jos kuningassivustan torni on
      liikkunut.
00063     bool _onkoMustaDTliikkunut;         // Linnoitus ei ole sallittu, jos daamisuvustan torni on
      liikkunut.
00064     bool _onkoMustaKTliikkunut;         // Linnoitus ei ole sallittu, jos kuningassivustan torni on
      liikkunut.
00065
00066     int mustienUpseerienLkm;
00067     int valkeidenUpseerienLkm;
00068     bool valkeaDaami;
00069     bool mustaDaami;
00070
00071     double laskeNappuloidenArvo(int);
00072     bool onkoAvausTaiKeskipeli(int);
00073     double nappuloitaKeskella(int);
00074     double linjat(int);
00075     bool onkoRuutuUhattu(Ruutu*, int vastustajanVari);
00076     void annaLinnoitusSiirrot(std::list<Siirto>& lista, int vari);
00077
00078     double ratsujaReunoilla(int vari);
00079     double sotilaat(int vari);
00080
00081
00082     // Karsii siirrot, jotka j♦tt♦v♦t oman K:n shakkiin.
00083     void huolehdiKuninkaanShakeista(std::list<Siirto>& lista, int vari);
00084 };
```

5.4 shakki/kayttoliittyma.cpp File Reference

```
#include "kayttoliittyma.h"
#include <fcntl.h>
#include <iostream>
#include <string>
```

5.5 shakki/kayttoliittyma.h File Reference

```
#include "asema.h"
#include "nappula.h"
#include "peli.h"
#include "siirto.h"
```

Classes

- class Kayttoliittyma

5.6 kayttoliittyma.h

Go to the documentation of this file.

```
00001 #pragma once
00002 #include "asema.h"
00003 #include "nappula.h"
00004 #include "peli.h"
00005 #include "siirto.h"
00006
00007 // Shakkiohjelman k�ytt�liittym�, joka osaa visualisoida nykyisen aseman
00008 // ja lukea k�ytt�j�n sy�tt�m�t siirrot. Singleton.
00009 class Kayttoliittyma
00010 {
00011 public:
00012     void aseta_asema(Asema *asema) { this->_asema = asema; }
00013     void piirraLauta(std::list<Siirto> &lista);
00014     Siirto annaVastustajanSiirto(std::list<Siirto>& lista);
00015     int kysyVastustajanVari();
00016
00017     static Kayttoliittyma *getInstance();
00018
00019 private:
00020     Asema *_asema;
00021     static Kayttoliittyma *instance; // osoitin luokan ainoaan olioon (Singleton).
00022
00023     Kayttoliittyma() {}
00024     Kayttoliittyma(Asema *asema) { this->_asema = asema; }
00025 };
```

5.7 shakki/main.cpp File Reference

```
#include <fcntl.h>
#include <iostream>
#include <string>
#include "asema.h"
#include "kayttoliittyma.h"
#include "siirto.h"
```

Functions

- int main ()

  Main function.

5.7.1 Function Documentation

5.7.1.1 main()

```
int main ( )
```

Main function.

Returns

int

5.8 shakki/minmaxpaluu.h File Reference

```
#include "siirto.h"
```

Classes

- class MinMaxPaluu

5.9 minmaxpaluu.h

Go to the documentation of this file.

```
00001 #pragma once
00002 #include "siirto.h"
00003
00004
00005 // luokka, jonka avulla saadaan palautettua minmax:ssa sekiirto-olio ettvaluointifunktion arvo
00006 // Struct ajaisi saman asian. Kun ei rakenneta gettereita settereitniin ei tarvita toteutus .cpp
      tiedostoa
00007 class MinMaxPaluu{
00008 public:
00009     double _evaluointiArvo;
00010     Siirto _parasSiirto;
00011 };
```

5.10 shakki/nappula.cpp File Reference

```
#include <list>
#include <string>
#include "asema.h"
#include "nappula.h"
```

5.11 shakki/nappula.h File Reference

```
#include <list>
#include <string>
#include "asema.h"
#include "siirto.h"
```

Classes

- class Nappula
- class Torni
- class Ratsu
- class Lahetti
- class Daami
- class Kuningas
- class Sotilas

Enumerations

- enum {
  VT , VR , VL , VD ,
  VK , VS , MT , MR ,
  ML , MD , MK , MS }

5.11.1 Enumeration Type Documentation

5.11.1.1 anonymous enum

```
anonymous enum
```

Enumerator

VT	
VR	
VL	
VD	
VK	
VS	
MT	
MR	
ML	
MD	
MK	
MS	

5.12 nappula.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include <list>
00004 #include <string>
00005 #include "asema.h"
00006 #include "siirto.h"
00007
00008 // Vakioarvot nappulatyypeille.
00009 enum
00010 {
00011     VT,
00012     VR,
00013     VL,
00014     VD,
00015     VK,
00016     VS,
00017     MT,
00018     MR,
00019     ML,
00020     MD,
00021     MK,
00022     MS
00023 };
00024
00025 // Yliluokka shakkinappuloille.
00026 class Nappula
00027 {
00028
00029 private:
00030     std::wstring _unicode; // nappulaa vastaava unicode-merkki
00031     int _vari;             // valkea = 0, musta = 1
00032     int _koodi;            // VT, VR, MT tms.
00033
00034 public:
00035     Nappula(std::wstring, int, int);
00036     Nappula() {}
00037
00038     // Siirtojen generointi. Puhdas virtuaalifunktio, eli aliluokat toteuttavat t♦m♦n
00039     // omalla tavallaan.
00040     virtual void annaSiirrot(std::list<Siirto> &lista, Ruutu *, Asema *, int vari) = 0;
00041
00042     void setUnicode(std::wstring unicode) { _unicode = unicode; }
00043     std::wstring getUnicode() { return _unicode; }
00044     void setVari(int vari) { _vari = vari; }
00045     int getVari() { return _vari; }
00046     int getKoodi() { return _koodi; }
00047     void setKoodi(int koodi) { _koodi = koodi; }
00048
00049     void siirrotSuuntaan(std::list<Siirto> &lista, Ruutu *, Asema *, int vari, int dx, int dy, int
      askeleet);
00050     void lisaaSotilaanKorotukset(Siirto*, std::list<Siirto>& lista, Asema*);
00051 };
00052
00053 // Torni-aliluokka. Virtuaalinen perint♦ tarkoittaa, ett♦ kantaluokka perit♦♦n moniperinn♦ss♦ vain
      kerran
00054 // (koska daami perii sek♦ tornin ett♦ l♦hetin).
00055 class Torni : public virtual Nappula
00056 {
00057 public:
00058     Torni(std::wstring unicode, int vari, int koodi) : Nappula(unicode, vari, koodi) {}
00059     void annaSiirrot(std::list<Siirto> &lista, Ruutu *, Asema *, int vari);
00060 };
00061
00062 // Ratsu-aliluokka.
00063 class Ratsu : public Nappula
00064 {
00065 public:
00066     Ratsu(std::wstring unicode, int vari, int koodi) : Nappula(unicode, vari, koodi) {}
00067     void annaSiirrot(std::list<Siirto> &lista, Ruutu *, Asema *, int vari);
00068 };
00069
00070 // L♦hetti-aliluokka. Virtuaalinen perint♦ tarkoittaa, ett♦ kantaluokka perit♦♦n moniperinn♦ss♦ vain
      kerran
00071 // (koska daami perii sek♦ tornin ett♦ l♦hetin).
00072 class Lahetti : public virtual Nappula
00073 {
00074 public:
00075     Lahetti(std::wstring unicode, int vari, int koodi) : Nappula(unicode, vari, koodi) {}
00076     void annaSiirrot(std::list<Siirto> &lista, Ruutu *, Asema *, int vari);
00077 };
00078
00079 // Daami-aliluokka. Perii sek♦ l♦hetin ett♦ tornin.
```

```
00080 class Daami : public Lahetti, public Torni
00081 {
00082 public:
00083     Daami(std::wstring unicode, int vari, int koodi) : Nappula(unicode, vari, koodi), Lahetti(unicode,
      vari, koodi), Torni(unicode, vari, koodi) {}
00084     void annaSiirrot(std::list<Siirto> &lista, Ruutu *, Asema *, int vari);
00085 };
00086
00087 // Kuningas-aliluokka.
00088 class Kuningas : public Nappula
00089 {
00090 public:
00091     Kuningas(std::wstring unicode, int vari, int koodi) : Nappula(unicode, vari, koodi) {}
00092     void annaSiirrot(std::list<Siirto> &lista, Ruutu *, Asema *, int vari);
00093 };
00094
00095 // Sotilas-aliluokka.
00096 class Sotilas : public Nappula
00097 {
00098 public:
00099     Sotilas(std::wstring unicode, int vari, int koodi) : Nappula(unicode, vari, koodi) {}
00100     void annaSiirrot(std::list<Siirto> &lista, Ruutu *, Asema *, int vari);
00101 };
```

5.13 shakki/peli.cpp File Reference

```
#include "peli.h"
```

5.14 shakki/peli.h File Reference

Classes

- class Peli

5.15 peli.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003
00004 // Peli tietokonetta vastaan joko mustilla tai valkeilla.
00005 class Peli
00006 {
00007 public:
00008     Peli(int);
00009     int getKoneenVari();
00010
00011 private:
00012     int _koneenVari; // Valkoinen = 0, Musta = 1
00013 };
```

5.16 shakki/ruutu.cpp File Reference

```
#include "ruutu.h"
```

5.17 shakki/ruutu.h File Reference

Classes

- class Ruutu

5.18 ruutu.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 // Shakkilaudan ruutu tietyss♦ (rivi, sarake) -koordinaatissa.
00004 class Ruutu
00005 {
00006 public:
00007     Ruutu(int, int);
00008     Ruutu();
00009
00010     bool operator==(const Ruutu &) const;
00011
00012     int getRivi();
00013     int getSarake();
00014     void setRivi(int);
00015     void setSarake(int);
00016
00017 private:
00018     int _sarake;
00019     int _rivi;
00020 };
```

5.19 shakki/siirto.cpp File Reference

```
#include "siirto.h"
```

5.20 shakki/siirto.h File Reference

```
#include "ruutu.h"
```

Classes

- class Siirto

5.21 siirto.h

Go to the documentation of this file.

```
00001 #pragma once
00002 #include "ruutu.h"
00003
00004 // Ns. "forward declaration". Nyt Asema-luokassa voidaa esitell\x94 Nappula-osoittimia ilman,
00005 // ett\x94 nappula.h -tiedostoa t\x94ytyy includoida.
00006 class Nappula;
00007
00008 // Siirto kuvaa nappulan siirtymisen ruudusta toiseen, mukaanlukien erikoissiirrot
00009 // (linnoitus ja ohestaly\x94nti).
00010 class Siirto
00011 {
00012 public:
00013     Siirto(Ruutu, Ruutu);
00014     Siirto();
00015     Siirto(bool, bool); // Linnoitus lyhesti (K-siipi) tai pitk\x94sti (D-siipi?
00016
00017     bool operator==(const Siirto &) const;
00018
00019     Ruutu getAlkuruutu();
00020     Ruutu getLoppuruutu();
00021     bool onkoLyhytLinna();
00022     bool onkoPitkalinna();
00023     Nappula *_miksikorotetaan = 0;
00024     void setAlkuruutu(Ruutu);
00025     void setLoppuruutu(Ruutu);
00026
00027 private:
00028     Ruutu _alkuRuutu;
00029     Ruutu _loppuRuutu;
00030     bool _lyhytLinna;
00031     bool _pitkaLinna;
00032 };
```

5.22 shakki/vastustaja.cpp File Reference

```
#include "vastustaja.h"
```

5.23 shakki/vastustaja.h File Reference

```
#include <string>
```

Classes

- class Vastustaja

5.24 vastustaja.h

Go to the documentation of this file.

```
00001 #pragma once
00002 #include <string>
00003
00004
00005 // Vastustajan tiedot.
00006 class Vastustaja
00007 {
00008 public:
00009     Vastustaja(std::wstring);
00010     std::wstring getNimimerkki();
00011
00012 private:
00013     std::wstring _nimimerkki;
00014 };
```